**4. CODING**

**4.1. Description of Software Used**

**Microsoft.net**

.NET is an integral part of many applications running on Windows and provides common functionality for those applications to run.Microsoft.NET is prefabricated infrastructure for solving common problems in internet applications. Microsoft.NET has been getting enormous amount of publicity lately, even for this industry Microsoft. NET is an add-on-run-time environment that runs on windows 2000 operating system. Later versions of .NET will probably be made part of the operating system, the U.S. department of justice willing.

**Microsoft.net framework**

The .NET Framework includes the runtime and compiles time services required to run a .NET application. Compile time is when the developer is compiling the source. Runtime is when the compiled code is executing in memory. At the center of the runtime execution of .NET code is the Common Language Runtime (CLR). The CLR is a virtual machine that runs as a process on the computer on which it is installed

**COMMON LANGUAGE RUNTIME**

The Common Language Runtime (CLR) serves as the execution environment for the .NET Framework. The CLR is responsible for managing the compiled code of .NET applications, which can be written in different languages including VB, C#, Java, and Perl. The cross-language integration is achieved through the two major components of CLR: Intermediate Language and Metadata.

Intermediate Language (IL) is an assembly language that runs on almost any type of CPU. IL achieves this versatility by using stacks to handle

functions that normally occur in registers. As managed code, IL is just-in-time (JIT) compiled when .NET applications are executed. JIT compilers convert IL into machine language that is specific to the host CPU. [During runtime, JIT compilers have the luxury of choosing stacks, registers or other stores to implement IL stacks.] Various JIT compilers are provided by the CLR, making it possible for different computer architectures to execute IL. Unlike other assembly languages, IL integrates high-level concepts which make CLR code more robust. As a high level language, IL is strongly typed and uses the ideas behind structured-exception handling, deployment support, and component interaction. Thus a range of software can run on the .NET Framework as long as the compiler can produce IL. Metadata, the second component of the CLR, is a description of the implemented code. The Metadata is responsible for ensuring that the CLR executes the code securely. To prevent modules of software from breaking type definitions, Metadata stores information regarding classes, methods, and types. Registers are no longer required to keep track of information because relevant data is stored with the compiled code or IL. By housing information on classes and registrations, Metadata allows the CLR to function more efficiently since programs are less likely to get hung up on version and inheritance dependencies.

**Language(c#)**

Microsoft Visual C# is a powerful but simple language aimed primarily at developers creating Applications by using the Microsoft .NET Framework. It inherits many of the best features of C++ and Microsoft Visual Basic, but few of the inconsistencies and anachronisms, resulting in a cleaner and more logical language. The advent of C# 3.0 has seen several important new features added to the language, including Generics, Iterates, and anonymous methods. By design, C# is the programming language that most directly reflects the underlying CLI.

Most of its intrinsic types correspond to value-types implemented by the Common Language Infrastructure (CLI) framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate Common Intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN

**Features of c#**

• C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.

• The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.

• The language is intended for use in developing software components

Suitable for deployment in distributed environments.

• Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.

Support for internationalization is very important.

• C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended

to compete directly on performance and size with C or assembly language.

**Microsoft SQL Server 2008**

Microsoft SQL Server 2008 provides the Microsoft Windows Server System integrated server software with a database platform for the next generation of connected, scalable, and reliable enterprise applications. The breadth and depth of innovation in this version is in response to t he needs of customers. This white paper is targeted to database administrators, to give you an understanding of the new features in and capabilities of SQL Server 2005. From many enhancement of existing features, to an entirely new security model, database administrative is now more productive and in tune with needs of the administrator.

At the core of SQL Server 2008 are new infrastructure application capabilities. SQL Service Broker is a distributed application framework that provides a new form of scalability and reliability for asynchronous message delivery. Though not new, Microsoft SQL server Notification Services, Reporting Services, and SQL Server Mobile Edition (formerly called SQL Server CE) are all greatly enhanced in SQL Server 2005.

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database scheme creation and modification, and database object access control management.

**4.2 Coding Principle**

The input to the coding phase is the design document. During coding phase, modules identified in the design document are coded according to the module specification. Objectives of coding phase are, to transform design into code and unit test the code.

**Coding Guidelines**

Code should be easy to understand.

Don’t take pride in cryptic code.

Code should be well documented.

Comments should be present.

Functions should be small.

Do not use Go-to statement.

Do not use the same variable for multiple purposes.